CLASSE DE ESTRUTURAS

"NØR"

CAUTL
ANTONIO GOUVEA PORTELA
LISBOA, NOVEMBRO 1982

INDEX

A) INTRODUÇÃO

Os programas da classe "NØR" têm por objectivo evitar a construção de programas "ad hoc" e poderem "emular" uma vasta classe de programas. Essencialmente constam de duas partes:

I Parte: Programa de interacção Homem-Computador, que permite ao primeiro não só introduzir dados, parâmetros, etc. mas "estruturar" o programa "NØR" em termos deste emulador num dado programa.

II Parte: Consta de um conjunto de sub-rotinas que podem classificar-se em três classes.

Classe 1: Sub-rotinas cíclicas isto é, sub-rotinas que deverão ser repetidas um número n de vezes. Note-se que se n=1 ou ∅ então a sub-rotina é executada uma só vez.

Classe 2: Sub-rotinas não cíclicas destinadas à execução de cálculos ou operações lógicas.

Classe 3: Sub-rotinas de "inter-face", isto é, que estabelecem a correspondência entre variáveis e destinadas a "ligar" duas sub-rotinas das classes 2 ou 1.

No caso particular do programa "NØR" foram instaladas: 5 sub-rotinas da classe 1 e prevista a sua expansão até 14.
5 Sub-rotinas de classe 2
3 sub-rotinas da classe 3

Previstas um número ilimitado de sub-rotinas das classes 2 a 3.

Nota 1 Todas as sub-rotinas da classe 1 têm a mesma estrutura e distinguem-se entre si:

a) por terem um número identificador P (1 a 5, podendo ser estendido a 14).

b) As variáveis da forma:

$$\left.\begin{array}{l} T(P,\alpha) \\ S(P,T(P,\alpha)) \\ O(P) \end{array}\right\} \quad \text{onde P é o número identificador}$$

permitem distinguir os vários ciclos entre si.

c) P=1 representa a sub-rotina <u>inicial</u>, as restantes ocuparão posições de sub-rotinas do ciclo P=1.

Nota 2: A distinção entre a classe 2 e 3 é mais conceptual e organizacional do do que uma necessidade, se for desejável que todas as sub-rotinas da classe 2 tomem a forma de formas paramétricas podendo servir vários problemas então convém distinguir a classe 2 da 3, por exemplo:

a) = sen x , poderá eventualmente ser substituído pela forma seguinte:

b) $= R_1+R_2 \sin(R_3 . x + R_4)$

A forma (b) inclui a forma (a), como coisa particular e tem a van-vantagem de poder tratar uma vasta classe de problemas, pelo que bastará acertar os parâmetros R_1 R_2 R_3 R_4.

c) =R+R sin (R +R+R)

Esta forma é equivalente à forma (b) porém com a vantagem de tratar R_1 R_2 R_3 e P_4 como variáveis.

A forma (b)

A forma (c)

No programa "NØR" todas as formas da classe 2 procuram formar formas semelhantes a (c).

Nota 3: Caso a classe 2 seja constituída essencialmente por formas generalizadas do tipo (c) então convirá que as variáveis R_i se distinguem de outras variáveis R_i pertencentes a outras sub-rotinas da classe 2 e por isso foi adoptada a simbologia seguinte:

$R(T,\alpha)$ onde: $\begin{cases} T \text{ é o número de identificação da sub-rotina da classe 2} \\ \alpha \text{ 2 um número de ordem arbitrário.} \end{cases}$

Por exemplo:

A expressão (c) $Y = R_1+R_2 \sin[R_3 * R_4+R_5]$ se pertencesse à sub-rotina identificada pelo Nº 18, escrever-se-ia como segue:

$$Y(18) = R(18,1) + R(18,2) \sin [R(18,3) \times R(18,4) + R(18,5)]$$

ou T=18 é o Nº identificador da sub-rotina e α=1, 2,...,5 é o número de ordem por que entram da esquerda para a direita as variáveis (e ou parâmetros) R (T,α).

Este procedimento reduz muito o Nº Sub-Rot. a escrever nos programas da classe "NØR".

Nota 4: O procedimento referido anteriormente quanto à simbologia das sub-rotinas da classe 2, implica então a necessidade de criar sub-rotinas da classe 3 (inter-faces) que realizam a conexão entre duas sub-rotinas de classe 2 que se sucedem; por exemplo:

$$S.R.18 \Rightarrow Y(18) = R(18,1) + R(18,2) \times \sin [R(18,3) \times R(18,4) + R(18,5)]$$

$$S.R.35 \Rightarrow Y(35) = R(35,1) + R(35,2) \times [R(35,3)]^2$$

e suponhamos que são dados.

$$S.R.18 \begin{cases} R(18,1) = 10 \\ R(18,2) = 1 \\ R(18,3) = \frac{\pi}{2} \\ R(18,5) = \pi \end{cases} \qquad S.R.35 \begin{cases} R(35,1) = 100 \\ R(35,2) = 4 \end{cases}$$

Então será necessário criar duas S.R. da classe 3, uma que antecede SR 18 e que consiste em $R(18,4)=Y(15)$ e outra entre SR.Y.18 e SR35 que consiste em $R(35,3)=Y(18)$ e a ordem de execução das sub-rotinas será:

SR 24 $\longrightarrow R(18,4) = Y(15)$
SR 18 $\longrightarrow Y(18) = 10 + 1 \times \sin\left[\frac{\pi}{2} \times R(18,4) + \pi\right]$
SR 25 $\longrightarrow R(35,3) = Y(18)$
SR 35 $\longrightarrow Y(35) = 100 + 4 \times \left[R(35,3)\right]^2$

Note-se que os restantes $R(18,\alpha)$ e $R(35,\beta)$, já tinham sido fornecidos ao serem introduzidos os dados.

Nota 5: Uma vez introduzidas todas as operações básicas e estas "generalidades (como se fez com o SIN no exemplo apresentado nas notas 2,3,4 o único tipo de programas (sub-rotina) a construir em cada caso pertence à classe 3, isto é, interfaces da forma $\{R=Y\}$

Nota 6: Esta sistematização das sub-rotinas em classes:

S.R Cíclicas: $\longrightarrow$ classe 1
S.R Operadoras $\longrightarrow$ " 2
S.R Interfaces $\longrightarrow$ " 3.

não impede que se construam S.R. para um fim determinado e específico, basta que se lhe associe um Nº identificador para a poder "chamar" quando for requerida.

Identicamente se podem dar números identificadores a <u>frases literais</u> ou a <u>funções</u>, definidas para determinado fim.

B) <u>Como Programar</u>

Um programa consta, na sua forma mais geral, de um conjunto de sub rotinas cíclicas e outras que se não repetem de cada vez que são "invocadas" várias vezes.

B1) Uma Sub-Rotinas cíclicas (S.R.C.)

Uma S.R.C consta de:

a) M nó de início

b) Uma sucessão de S.R ($\alpha, \beta, \ldots, \delta$)

c) D dum nó final onde está um decisor que de acordo com instruções recebidas previamente opta entre voltar ao nó inícial M e dar a SRC por concluída (S)

Fig: 1

Invocada uma dada S.R.C, alguns dados são fornecidos à entrada E, nomeadamente:

i) Quantos ciclos devem ser feitos, para que D saiba optar entre repetir ou terminar (S).

e) Quantas sub-rotinas terão de ser executadas e os números de referência respectivas dados pela ordem em que essas S.R. terão de ser executadas:

Na figura 1 vê-se que as sub-rotinas a executar são por ordem de execução:

SR$\alpha \rightarrow$ SR$\beta \rightarrow \ldots \rightarrow$ S.R.

Repare-se que as S.R.C. não executam cálculos ou operações lógicas, limitam-se a "invocar" como S.R., as sub-rotinas que directa ou indirectamente as executam.

A execução indirecta corresponde aos casos onde a S.R. invocada é também cíclica porque então será esta directa ou indirectamente que as executa, veja-se Fig.2.

O número de S.R.C que podem, em série, invocar é, em geral, 5 a 6 e depende do software básico onde o programa da classe "NØR" foi implantado.

Note-se que o número de S.R. cíclicas que podem ser invocadas não está limitado aos 5 ou 6 atrás referidos.

Para o efeito veja-se o esquema apresentado na fig.3.

Para o efeito de representação gráfica as sub-rotinas cíclicas são representadas por círculos, as restantes por rectângulos.

Como se vê a SRC principal (1) invoca três SRC.

A SRC de nível (2) invocada em primeiro lugar, invoca por seu turno uma SRC de nível 3 e assim sucessivamente até ao nível (5 ou 6).

Formas periódicas discontínuas anti-simétricas

1) INTRODUÇÃO

Seja $z(\alpha) = R(\alpha) \times \exp(i\alpha)$ um complexo satisfazendo as duas seguintes condições:

$$(1) \quad \begin{cases} A)\ R(\alpha + 2\pi) = R(\alpha) \\ \\ B)\ R(2\pi - \alpha) = R(\alpha) \end{cases}$$

(Periódico em α) e $R(\alpha)$ é mera aplicação.

onde R é um real positivo e finito e α um real.

De (1) resulta:

$$(2) \text{ ------- } \begin{cases} z(\alpha + 2\pi) = z(\alpha) \\ \\ z(2\pi - \alpha) = R(\alpha) \times \exp[-i\alpha] = z^*(\alpha) \end{cases}$$

Seja $z_m(\alpha) = z\left(\alpha + \frac{2\pi m}{N}\right)$ ------------------ (3)

onde n, N $\in$ são inteiros não negativos e n $\leqslant$ N , será ainda:

$$z_m(\alpha) = R\left(\alpha + \frac{2\pi m}{N}\right) \times \exp\left[i\,\frac{2\pi m}{N}\right] \times \exp[i\alpha]$$

Finalmente seja: $\sigma_N = \sum_{m=0}^{N-1} z_m(\alpha)$ ------------- (4)

ou $\sigma_N = \exp[i\alpha] \times \sum_{m=0}^{N-1} R\left(\alpha + \frac{2\pi m}{N}\right) \times \exp\left[i\,\frac{2\pi m}{N}\right]$ (5)

O problema consiste em estudar σ_N conhecidos: $z(\alpha)$ e N .

No ponto 2) apresentam-se propriedades de σ_N , em 3) alguns desenvolvimentos, em 4) resultados, e, finalmente, em 5) programas que permitem calcular as diversas expressões.

Nota: onde (*) é o símbolo de conjugado e exp$[\ ] = e^{[\ ]}$.

2) PROPRIEDADES PRINCIPAIS

Se $Z(\alpha)$ satisfizer a A) e B) então $\mathcal{Z}_N(\alpha)$ também satisfaz.

Com efeito:

$$\mathcal{Z}_N(\alpha+2\pi) = \exp\left[i(\alpha+2\pi)\right] \sum_{m=0}^{N-1} R\left(\alpha+2\pi+\frac{2\pi m}{N}\right) \times \exp\left[i\,\frac{2\pi m}{N}\right]$$

e atendendo a (1) temos:

$$= \exp\left[i\alpha\right] \sum_{m=0}^{N-1} R\left(\alpha+\frac{2\pi m}{N}\right) \times \exp\left[i\,\frac{2\pi m}{N}\right]$$

$$= \mathcal{Z}_N(\alpha)$$

(veja-se 5)

$$\mathcal{Z}_N(2\pi-\alpha) = \sum_{m=0}^{N-1} Z_m\left(2\pi-\alpha+\frac{2\pi m}{N}\right)$$

(veja-se 4)

mas $Z_m\left(2\pi-\alpha+\frac{2\pi m}{N}\right) = Z_m^*\left(\alpha-\frac{2\pi m}{N}\right)$ (veja-se 2)

e ainda $= Z_m^*\left(\alpha+\frac{2\pi(N-m)}{N}\right)$

e fazendo $K = N-m$ e substituindo em $\mathcal{Z}_N$ temos:

$$\mathcal{Z}_N(2\pi-\alpha) = \sum_{K=0}^{N-1} Z_m^* \frac{2\pi K}{N}$$

$$= \mathcal{Z}_N^*(\alpha)$$

q.e.d.

Assim as propriedades A e B são conservadas em $\mathcal{Z}_N$.

3) ALGUNS DESENVOLVIMENTOS

A expressão $\mathcal{Z}_N$ pode tomar a forma desenvolvida seguinte:

(6) $$\mathcal{Z}_N(\alpha) = (\cos\alpha + i\sin\alpha) \times \sum_{m=0}^{N-1} R\left(\alpha+\frac{2\pi m}{N}\right)\left(\cos\frac{2\pi m}{N} + i\sin\frac{2\pi m}{N}\right)$$

e seja:

(7) $$F_N(\alpha) = \sum_{m=0}^{N-1} R\left(\alpha+\frac{2\pi m}{N}\right) \times \cos\frac{2\pi m}{N}$$

(8) $$G_N(\alpha) = \sum_{m=0}^{N-1} R\left(\alpha + \frac{2\pi m}{N}\right) \times \sin \frac{2\pi m}{N}$$

Propriedades de F_N e G_N

a) $F_N(\alpha + 2\pi) = F_N(\alpha)$ e $G_N(\alpha + 2\pi) = G_N(\alpha)$

é evidente a sua prova.

b) $$F_N(2\pi - \alpha) = \sum_{m=0}^{N-1} R\left(2\pi - \alpha - \frac{2\pi m}{N}\right) \cos \frac{2\pi m}{N}$$

$$= \sum_{m=0}^{N-1} R\left(\alpha + \frac{2\pi m}{N}\right) \times \cos \frac{2\pi m}{N}$$

mas

$$\cos\left(\frac{2\pi m}{N}\right) = \cos \frac{2\pi (N-m)}{N}$$

Donde (9)

$$F_N(2\pi - \alpha) = F_N(\alpha) \ldots$$

e

$$G_N(2\pi - \alpha) = \sum_{m=0}^{N-1} R\left(\alpha + \frac{2\pi m}{N}\right) \times \sin \frac{2\pi m}{N}$$

mas sim

$$\left(\frac{2\pi m}{N}\right) = -\sin \frac{2\pi (N-m)}{N}$$

e

$$G_N(2\pi - \alpha) = -G_N(\alpha) \ldots$$

F_N é simétrico

G_N é anti-simétrico

Finalmente

$$Z_N(2\pi - \alpha) = \left[\cos(2\pi - \alpha) + i \sin(2\pi - \alpha)\right] \times$$

$$\times \left[F_N(2\pi - \alpha) + i\, G_N(2\pi - \alpha)\right]$$

$$= \left[\cos \alpha - i \sin \alpha\right] \times \left[F_N(\alpha) - i\, G_N(\alpha)\right]$$

$$= Z_N^*(\alpha)$$

(Regra do produto de conjugados)

4) RESULTADOS FINAIS

a) Nenhuma restrição foi feita a $R(\alpha)$ a não ser que não seja ∞ , note-se que pode ser descontínuo.

b) $F_N(\alpha)$ e $G_N(\alpha)$, conhecido $Z(\alpha)$, e escolhido N podem ser calculados, uma vez por todas, no intervalo $\alpha \in (0, 2\pi]$.

c) Toda a informação reduz-se ao conhecimento de $\underline{\underline{F_N}}$, $\underline{\underline{G_n}}$ e N.

d) Os programas que se seguem destinam-se a efectuar esses cálculos.

e) Reproduzem-se F_N, G_N e $\mathcal{Z}_N$.

$$F_N(\alpha) = \sum_{m=0}^{N-1} R\left(\alpha + \frac{2\pi m}{N}\right) \cos \frac{2\pi m}{N}$$

$$G_N(\alpha) = \sum_{m=0}^{N-1} R\left(\alpha + \frac{2\pi m}{N}\right) \sin \frac{2\pi m}{N}$$

$$\mathcal{Z}_N(\alpha) = \left[\cos\alpha + i \sin\alpha\right] \times \left[F_N(\alpha) + i\, G_N(\alpha)\right]$$

5) PROGRAMAS

PROGRAMA "TRIC"

Calcula:

$$(P) \Rightarrow F_N(\alpha) = \sum_{m=0}^{N-1} R\left(\alpha + \frac{2\pi m}{N}\right) \cos \frac{2\pi m}{N}$$

$$(\varphi) \Rightarrow G_N(\alpha) = \sum_{m=0}^{N-1} R\left(\alpha + \frac{2\pi m}{N}\right) \sin \frac{2\pi m}{N}$$

$$(L) \Rightarrow \varphi \times \sin\alpha = G_N \sin\alpha$$

$$(M) \Rightarrow P \times \cos\alpha = F_N \cos\alpha$$

$$(U) \Rightarrow \varphi \times \cos\alpha = G_N \cos\alpha$$

$$(V) \Rightarrow P \times \sin\alpha = F_N \sin\alpha$$

INSTRUCÇÕES DE PLOT E PRINT

	12	13	14
E1	Q	L	U
E2	P	M	V

$\leftarrow$ PLOT

GRÁFICO

KØ = deslocamento vert.

K = amplificação

E $\rightarrow$

$\geqslant$ 20	$\geqslant$ 30	$\geqslant$ 40
Q e P	0	STOP

$\leftarrow$ PRINT

OUTRAS INSTRUCÇÕES

NUM.MAX.de FASES (F8), por exemplo 30.

NUM. de FASES (F9) $\quad F9 < F8$

NUM.SALTOS (N9)(N) $\quad \alpha_{i+1} = \alpha_i + d\alpha$

$N \times d\alpha = 2\pi$, por exemplo: 120 ou 200.

Da 1ª vez indicar RUN. Para repetições basta indicar GOTO 200 e dar o nº de fases.

PROGRAMA "TRIC"

"TRICONT"

```
                                    FIM

  10 REM "TRIC"
  15 LET E=0
  20 PRINT "ESCREV,(Q=E1,P=E2: 1
2) (L=E1,M=E2: 13) (U=E1,V=E2:14
)"
  22 INPUT E1
  25 PRINT E1
  27 INPUT E2
  29 PRINT E2
  30 PRINT "DESLOC.VERTIC."
  35 INPUT K0
  36 PRINT K0
  37 LET K0=-K0
  45 PRINT "AMPLIFICACAO  K"
  46 INPUT K
  47 PRINT K
  50 PRINT "NUM.MAX.DE FASES  F8
"
  60 INPUT F8
  70 PRINT F8
  80 PRINT "NUMER.SALTOS N9"
  90 INPUT N9
 100 PRINT N9
 110 DIM L(N9+5)
 120 DIM M(N9+5)
 130 DIM U(N9+5)
 140 DIM V(N9+5)
 150 DIM A(F8+5)
 160 DIM S(F8+5)
 170 DIM C(F8+5)
 180 DIM P(N9+5)
 190 DIM Q(N9+5)
 200 PRINT "NUM.DE FASES F9?"
 205 INPUT F9
 206 PRINT "F9=";F9
 210 LET H=2*PI/F9
 220 FOR I=1 TO F9
 230 LET A(I)=H*(I)
 240 LET S(I)=SIN A(I)
 250 LET C(I)=COS A(I)
 260 NEXT I
 300 FOR J=0 TO N9-1
 310 LET Q=0
 320 LET P=0
 330 FOR I=1 TO F9
 340 GOSUB 1000
 350 LET Q=Q+R*S(I)
 360 LET P=P+R*C(I)
 370 NEXT I
 380 GOSUB 1500
 400 IF E>=20 THEN PRINT AT 0,0;
"QY";J;"=";Q;" PX";J;"=";P
 403 LET J5=J+1
 405 LET J3=60*J/N9
 407 GOSUB 1700
 410 IF E>40 THEN STOP
 420 NEXT J
 430 PRINT AT 20,25;"FIM"
 440 STOP
 450 GOTO 10
1000 REM "CALCULO FUNCAO"
1010 LET O=2*PI*J/N9+A(I)
1015 IF E>30 THEN PRINT "O=";O
1020 IF O>PI/2 THEN GOTO 1040
1030 LET R=3*O
1035 GOTO 1130
1040 IF O>[illegible] THEN GOTO 1060
1050 LET R=[illegible]
1055 GOTO [illegible]
1060 IF O>[illegible] GOTO 1080
1070 LET R=[illegible]
1075 GOTO 1[illegible]
1080 IF O>[illegible] THEN GOTO 1110
1090 LET R=[illegible]
1100 GOTO 113[illegible]
1110 LET [illegible]
1120 GOTO [illegible]
1130 IF [illegible]
1140 RET[illegible]
1150 STOP
1500 REM [illegible]
1510 LET [illegible]
1520 LET [illegible]
1530 LET [illegible]
1540 LET [illegible]
1550 LET [illegible] A1
1560 LET [illegible] A1
1570 LET [illegible] A1
1580 LET [illegible] A1
1590 RETURN
1700 REM "PLOT"
1730 IF E1=12 THEN PLOT J3,Q(J5)
*K+K0
1740 IF E[illegible] PLOT [illegible]
*K+K0
1750 IF [illegible]
5)+K0
1760 IF [illegible]
5)+K0
1770 IF E1=[illegible]
5)+K0
1780 IF [illegible]
5)+K0
1790 RET[illegible]
2000 REM [illegible]
2010 FOR [illegible]
2015 LET [illegible]
2020 PRINT [illegible]
2030 PRINT [illegible]
2040 PRINT [illegible]
2050 PRINT [illegible]
2060 NEXT [illegible]
2070 PRINT [illegible]
2080 STOP
```

As funções estudadas foram :

$\underline{y} = 1$

$R = 3 \times X \quad X \leq \frac{\Pi}{2} \qquad X = \hat{a}ngulo$

$R = 1/3 \times X \quad \frac{\pi}{2} < X \leq \pi$

$R = 1/3 \times X \quad \pi < X \leq \frac{3}{2}\pi$

$R = 3 \times X \quad \frac{3}{2}\pi < X \leq 2\pi$

Página 7 a 11.

e

$\underline{y} = 2$

R = arco sin x $\qquad \forall x$

Página 12.

A SRC de nível (2) invocada em segundo lugar limita-se a executar sub-rotinas não ciclicas

A SRC de nível (2) invocada em terceiro lugar invoca outra de nível (3) e esta outra do nível (4).

Finalmente, como se vê na fig.3, há:

1 SRC no 1º nível 3 SRC no 2º nível 2 SRC no 3º nível 2 SRC no 4º nível 1 SRC no 5º nível	ao todo 9 sub-rot cíclicas

Note-se que o programa "NØR3", pode suportar 14 sub-rotinas cíclicas, embora na presente configuração só estejam instaladas 5 S.R.C..

B2) Sub-Rotinas não cíclicas

Dividem-se em dois grandes grupos:

1º as que realizam operações lógicas ou numéricas usais.

2º as que constituem "interfaces," isto é, que se destinam a transferir valores suportados por certas variáveis e que é preciso transferir para outras variáveis.

Quanto às SR do tipo 1º nada há a dizer porque devem corresponder às S.R usuais suportadas no soft-ware básico instalado no computador, contudo nada impede que para fins específicos se acrescentem todas as S.R que se achar de utilidade.

Quanta às S.R do tipo 2º, limitam-se à formula

R( )=Y( )
R( )=R( )
Y( )=Y( )
Y( )=R( )

e podem ser suportadas em SRC próprias quando há que transferir matrizes ou vectores.

C) Exemplo

Para explicar os capítulos (A e B), convém apresentar um exemplo completo.

O problema consiste em calcular:

$$X(3, \alpha) = Y(\beta) + Y(\gamma)$$
$$Y(\beta) = R(\beta,1) + R(\beta,2) \times \mathrm{SIN}[R(\beta,3) * R(\beta,4) + R(\beta,5)]$$
$$Y(\gamma) = R(\gamma,1) + R(\gamma,2) \times \mathrm{ARCSIN}[R(\gamma,3) * R(\gamma,4) + R(\gamma,5)]$$

onde $\left\{ \begin{matrix} R(\beta,1), & R(\beta,2), & R(\beta,5) \\ R(\gamma,1), & R(\gamma,2), & R(\gamma,5) \end{matrix} \right\}$ são dados

$\left\{ \begin{matrix} R(\beta,3) \\ R(\gamma,3) \end{matrix} \right\}$ são parâmetros que tomam sucessivamente vários valores

$\left\{ \begin{matrix} R(\beta,4) \\ R(\gamma,4) \end{matrix} \right\}$ constituem a variável que também toma vários valores

Será assim necessário pelo menos <u>dois ciclos</u>, um para gerar Y a variável correspondente a $R(\beta,4)=R(\gamma,4)=\gamma$, um outro ciclo para gerar Y , correspondente aos parâmetros $R(\beta,3)=R(\gamma,3)=\varphi'$

Por exemplo: φ tomará valores: 1,2,...,30

φ' " " $R(\delta,1)+R\left[(\delta,2)-R(\delta,1)\right]*R(\delta,3)\div R(\delta,4)$

onde: $R(\delta,1)=\phi$, $R(\delta,2)=\pi/2$ e $R(\delta,4)=0{,}01$

e $R(\delta,3)=1,2,\ldots,10$

Passando a implantação deste programa:

<u>No ciclo 1</u> (S.R.C.1) vamos suportar a geração do parâmetro φ':

Número do ciclo : 1=P

Quantos ciclos 10=T(1,4)

Quantas S.R 3=T(1,2)

Número indicativo da 1ª S.R.=15

Número " " 2ª S.R.=16

" " " 3ª S.R.= 2 (S.R.C.2)

A <u>Sub-Rotina 15</u> $\begin{cases} R(16,4)=T(1,4) \\ R(16,3)=T(1,3) \end{cases}$

A <u>Sub-Rotina 16</u> $Y(16)=R(16,1)+\left(R(16,2)-R(16,1)\right)*$

$\delta=16$ $*R(16,3)\div R(16,4)$

A <u>Sub-Rotina 2</u> representa uma S.R.C. destinada a gerar φ.

No <u>Ciclo 2</u> (S.R.C.2) vamos suportar a geração do parâmetro φ.

Número do ciclo   2=P

Quantos ciclos 3=T(2,4)

Quantas sub-rotinas 6=T(2,2)

número	indicativo	da	1ª	S.R.	= 17
"	"	"	2ª	"	= 18
"	"	"	3ª	"	= 19
"	"	"	4ª	"	= 20
"	"	"	5ª	"	= 21
"	"	"	6ª	"	= 22

Quantos	parâmetros	têm	a	1ª	S.R.	=
"	"	"	"	2ª	"	= 5
"	"	"	"	3ª	"	=
"	"	"	"	4ª	"	=
"	"	"	"	5ª	"	=
"	"	"	"	6ª	"	= 6

A <u>Sub-Rotina 17</u>   R(18,3)=R(19,3)= y(16)- y(16) valor paramétrico calculado na S.R.C.1.

R(18,4)=R(19,4)=T(2,3) (valor de y)

<u>Sub-Rot 18</u>   y (18)=R(18,1)+R(18,2)xSIN[R(18,3)*R(18,4)+R(18,5)]

onde: R(18,1), R(18,2) e R(18,5) são dados, R(18,3) calculado SR 17

R(18,4)   "   SR 17

<u>Sub-Rot 19</u>   (19)=R(19,1)+R(19,2)x ARCSIN [R(19,3)xR(19,4)+R(19,5)]

onde: R(19,1),R(19,2) e R(19,5) são dados, R(19,3) e R(19,4) calculados R 17

<u>Sub-Rot. 20</u>   x(1,T(2,3)) = y(18)

x(2,T(2,3)) = y(19

x(3,T(2,3)) = y(18)= (19)

Sub-Rot. 22 R(21,3)=T(2,3) T(2,4)

R(21,6)=x(3,T(2,3))

Prepara parâmetro para o Ploting.

Sub-Rot. 21 Plot R(21,1)+R(21,2)*R(21,3)

R(21,4)+R(21,5)*R(21,6)

onde: R(21,1), R(21,2),R(21,4),R(21,5)são dados

e R(21,3) e R(21,6) são calculados em SR 22.

A execução inicia-se pelo S.R.C.1, que invoca a S.R.C.2 entre outras.

Este exemplo simples explica o método de programação.

D) Interface Homem-Computador

A forma como a "interface" se materializa é talvez a forma num modo simples de a escrever:

D1) Instruções Genéricas

1ª Questão "Quer Ins. Dados? (não = ϕ)"

Esta pergunta destina-se a saber se os dados já existem em memória ou em cassete e então não será necessário "injectar dados" e a resposta é [9] isto é, não é [ϕ]. A informação fica retida na variável Z 1.

1) Se $Z_1 \neq \phi$

2ª Questão "Quer regist? (Não=ϕ)"

Esta pergunta destina-se a saber se se quer registar os dados em cassete, por exemplo. Se se responder sim ≠ ϕ vai para 1.1,. A variável de suporte em Z2.

1.1) Se Z2 ≠ ϕ (e Z1 ≠ ϕ)

3ª Questão "Nome do registo"

O nome do registo é memorizado em Z$ e terá os símbolos que forem autorizados pelo soft-ware implantado no computador.

4ª Qustão "Escreva (Create. Z$, Número de registos, 12) e depois cont"

Trata-se de uma instrução para criar um ficheiro de dados cujo nome é Z$ que tem um certo número de registos (50 a 150 ;função do tamanho do programa e do número de parâmetros envolvidos), com 16 Byts por registo.

Esta instrução é executada do teclado e no fim carregar na tecla de execução.

Para o programa prosseguir é necessário carregar na tecla "CONT".

5ª Questão "Quer escrev. dados (não = ∅)"

Há medida que os dados são injectados pode ou não querer-se que os dados sejam escritos (Print) ou apresentados (display) e esta 5ª questão esclarece este ponto. A variável Z4 retém esta instrução de escrita.

Chegado a esta altura entra-se propriamente no questionário relativo ao programa específico, dados e instruções que se deseja implatar no programa da classe "NØR".

Retomaremos mais adiante em D2 a descrição.

1.2) Se $Z2=\phi$ (e $Z1\neq\phi$)

Então o programa de questões passa logo para a 5ª questão (instrução de escrita), e daí para D2.

Neste caso a informação fica registada apenas na memória central mas não na cassete.

2) Se $Z1=\phi$

6ª Questão "Quer ler gravador (não = ∅)"

O objectivo é saber se os dados necessitam de ser lidos do gravador (cassete) ou não.

Esta informação está suportada em Z3.